Notice of Allowability	Application No.	Applicant(s)	
	09/583,411	TAYLOR, KURT RUSSELL	
	Examiner	Art Unit	
	LeChi Truong	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *BPAI Decision on 01/262007*.

2. ☒ The allowed claim(s) is/are *2-4, 6, 8, 11, 14-19, 21-23, 25, 27, 30, 33-38, 40-42, 44, 46,49, 52-57now renumbered as claims 1-36*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Examiner's Amendment

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no lather than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with Mr. Gerald H. Glanzman (registration number: 25,035) on 04/20/2007.

3. Amend the following claims:

1. (Canceled)

2. (Currently Amended) The method of claim [[1]] 6, wherein the registry associated with the OID abstraction layer provides information identifying an anchor point in the OID subtree structure to be maintained by the repository.

3. (Original) The method of claim 2, wherein if the anchor point of the OID subtree structure is already registered with the OID abstraction layer, the registry is overwritten.

4. (Original) The method of claim 2, wherein if a query is received for an object that has an

Object Identifier that is below a registered anchor point in an OID tree structure, the OID abstraction layer identifies a repository that maintains object information for the requested object based on the registered anchor point.

5. (Canceled)

6. (Currently Amended) A method on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the method comprising the steps of:

receiving, in an OID abstraction layer, an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

registering the OID tree structure with a registry associated with the OID abstraction layer; and

adding the OID tree structure to a repository associated with the OID abstraction layer, wherein the repository is configured such that the repository recognizes requests from an application program interface (API) associated with the OID abstraction layer and sends reply messages to the API containing information retrieved from the repository, and ~~The method of claim 5,~~ wherein the OID abstraction layer receives the information retrieved from the repository through the API and encapsulates the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target

protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

7. (Canceled)

8. (Currently Amended) A method on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the method comprising the steps of:

receiving, in an OID abstraction layer, an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

registering the OID tree structure with a registry associated with the OID abstraction layer; and

adding the OID tree structure to a repository associated with the OID abstraction layer, wherein the OID abstraction layer receives a request for object data from a requesting protocol interface, interprets the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols, converts the request into an application program interface (API) request that is forwarded to the repository, and receives an API reply from the repository having the object data, and ~~The method of claim 7~~, wherein the OID abstraction layer reformats the object data in a

reply message according to the protocol of the requesting protocol interface and sends the reply message to the requesting protocol interface.

9-10. (Canceled)

11. (Currently Amended) The method of claim [[10]] 14, wherein if the first query cannot be mapped into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

12-13. (Canceled)

14. (Currently Amended) A method on a server in a distributed data processing system for retrieving object data from a repository, comprising:

receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer, wherein the first query is mapped into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer, wherein the second query is sent to the repository that contains the object associated with the first query; and

retrieving the object data from the repository using an OID abstraction layer application program interface (API), wherein a first reply is received at the API associated with the OID abstraction layer from the repository that contains the object associated with the first query, and ~~The method of claim 13,~~ wherein the first reply is transformed into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

15. (Original) The method of claim 14, wherein the second reply is sent to the requester in the distributed data processing system.

16. (Currently Amended) A method on a server in a distributed data processing system for retrieving object data from a repository, comprising:

receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or

more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

retrieving the object data from the repository using an OID abstraction layer application program interface (API) ~~The method of claim 9~~, wherein each repository in a plurality of repositories contains information representing an Object Identifier (OID) subtree structure, and wherein the plurality of repositories are formatted to support the two or more different protocols.

17. (Currently Amended) The method of claim [[9]] 16, wherein Simple Network Management Protocol (SNMP) is a protocol recognized by the OID abstraction layer.

18. (Currently Amended) The method of claim [[9]] 16, wherein Lightweight Directory Access Protocol (LDAP) is a protocol recognized by the OID abstraction layer.

19. (Currently Amended) The method of claim [[9]] 16, wherein Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) is a protocol recognized by the OID abstraction layer.

20. (Canceled)

21. (Currently Amended) The apparatus of claim [[20]] 25, wherein the registry provides information identifying an anchor point in the OID tree structure to be maintained by the repository.

22. (Original) The apparatus of claim 21, wherein if the anchor point of the OID tree structure is already registered in the registry, then the registry overwrites the previous entry.

23. (Original) The apparatus of claim 21, wherein, if the OID abstraction layer receives a query for an object that has an Object Identifier that is below a registered anchor point in an OID tree structure, the registry in the OID abstraction layer identifies a repository that maintains object information for the requested object based on the registered anchor point.

24. (Canceled)

25. (Currently Amended) An apparatus on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the apparatus comprising:

an OID abstraction layer that receives an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols

and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a registry, associated with the OID abstraction layer, that registers the OID tree structure; and

an adding means for adding the OID tree structure to a repository associated with the OID abstraction layer, wherein the repository is configured such that the repository recognizes requests received from an application program interface (API) associated with the OID abstraction layer and sends reply messages to the API containing information retrieved from the repository, and ~~The apparatus of claim 24,~~ wherein the OID abstraction layer receives the information retrieved from the repositories through the API and encapsulates the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

26. (Canceled)

27. (Currently Amended) An apparatus on a server in a distributed data processing system for maintaining a logical composite repository of Object Identifier (OID) tree structures, the apparatus comprising:

an OID abstraction layer that receives an OID tree structure from a repository; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a registry, associated with the OID abstraction layer, that registers the OID tree structure; and

an adding means for adding the OID tree structure to a repository associated with the OID abstraction layer, wherein the OID abstraction layer receives a request for object data from a requesting protocol interface, interprets the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols, converts the request into an application program interface (API) request that is forwarded to the repository, and receives an API reply from the repository having the object data, and ~~The apparatus of claim 26,~~ wherein the OID abstraction layer encapsulates the object data in a reply message according to the protocol of the requesting protocol interface and sends the reply message to the requesting protocol interface.

28-29. (Canceled)

30. (Currently Amended) The apparatus of claim [[29]] 33, wherein if the mapping means cannot map the first query into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

31-32. (Canceled)

33. (Currently Amended) An apparatus on a server in a distributed data processing system for retrieving object data from a repository, comprising:

a receiving means for receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a interpreting means for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

a mapping means for mapping the first query into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer;

a locating means for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer;

a first sending means, in the OID abstraction layer, that sends the second query to a repository that contains the object associated with the first query;

a retrieving means for retrieving the object data from the repository using an OID abstraction layer application program interface (API), wherein the retrieving means receives a first reply at the API from the repository that contains the object associated with the first query; and

~~The apparatus of claim 32, further comprising~~ a transforming means, in the OID abstraction layer, that transforms the first reply into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

34. (Original) The apparatus of claim 33, further comprising a second sending means, in the OID abstraction layer, that sends the second reply to the requester in the distributed data processing system.

35. (Currently Amended) An apparatus on a server in a distributed data processing system for retrieving object data from a repository, comprising:

a receiving means for receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

a interpreting means for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

a locating means for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

a retrieving means for retrieving the object data from the repository using an OID abstraction layer application program interface (API) ~~The apparatus of claim 28~~, wherein each repository in a plurality of repositories contains Object Identifier (OID) tree structures, and wherein the plurality of repositories are formatted to support the two or more different protocols.

36. (Currently Amended) The apparatus of claim [[28]] 35, wherein the receiving means recognizes a Simple Network Management Protocol (SNMP) query.

37. (Currently Amended) The apparatus of claim [[28]] 35, wherein the receiving means recognizes a Lightweight Directory Access Protocol (LDAP) query.

38. (Currently Amended) The apparatus of claim [[28]] 35, wherein the receiving means recognizes a Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) query.

39. (Canceled)

40. (Currently Amended) The computer program product of claim [[39]] 44, further comprising instructions for maintaining the registry associated with the OID abstraction layer

and providing information identifying an anchor point in the OID tree structure to be maintained by the repository.

41. (Original) The computer program product of claim 40, wherein if the anchor point of the OID tree structure is already registered with the OID abstraction layer, the instructions for registering overwrites the previous entry.

42. (Original) The computer program product of claim 40, further comprising instructions for identifying a repository that maintains object information for the requested object based on the registered anchor point if a query is received for an object that has an Object Identifier that is below a registered anchor point in an OID tree structure.

43. (Canceled)

44. (Currently Amended) A computer program product in a computer readable medium for maintaining a repository of Object Identifier (OID) tree structures, comprising:

instructions for receiving, in an OID abstraction layer, an OID tree structure from a repository;

wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for registering the OID tree structure with a registry associated with the OID abstraction layer;

instructions for adding the OID tree structure to a repository associated with the OID abstraction layer;

instructions for configuring the repository to recognize requests from an application program interface (API) associated with the OID abstraction layer and to send reply messages to the API containing information retrieved from the repository; and

~~The computer program product of claim 43, further comprising~~ instructions for receiving the information retrieved from the repository, through the API, and encapsulating the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols.

45. (Canceled)

46. (Currently Amended) A computer program product in a computer readable medium for maintaining a repository of Object Identifier (OID) tree structures, comprising:

instructions for receiving, in an OID abstraction layer, an OID tree structure from a repository;

wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for registering the OID tree structure with a registry associated with the OID abstraction layer;

instructions for adding the OID tree structure to a repository associated with the OID abstraction layer;

instructions for receiving a request for object data from a requesting protocol interface;

instructions for interpreting the request according to a protocol of the requesting protocol interface, wherein the protocol of the requesting protocol interface is one of the two or more different protocols;

instructions for converting the request into an application program interface (API) request which is forwarded to the subtree repository;

instructions for receiving an API reply from the subtree repository having the object data; and

~~The computer program product of claim 45, further comprising~~ instructions for encapsulating the object data in a reply message according to the protocol of the requesting protocol interface and sending the reply message to the requesting protocol interface.

47-48. (Canceled)

49. (Currently Amended) The computer program product of claim [[47]] 52, wherein if the instructions for receiving the first query map cannot map the first query into a second query due to a limitation of the repository that contains the object associated with the first query, then the first query cannot be satisfied.

50-51. (Canceled)

52. (Currently Amended) A computer program product in a computer readable medium for retrieving object data from a repository, comprising:

instructions for receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for mapping the first query into a second query, wherein the second query is consistent with an application program interface (API) associated with the OID abstraction layer;

instructions for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

instructions for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer;

instructions for sending the second query to the repository that contains the object associated with the first query;

instructions for retrieving the object data from the repository using an OID abstraction layer application program interface (API);

instructions for receiving a first reply at the API associated with the OID abstraction layer from the repository that contains the object associated with the first query; and

~~The computer program product of claim 51, further comprising~~ instructions for transforming the first reply into a second reply, wherein the second reply is consistent with the protocol for the first query recognized by the OID abstraction layer, and wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols.

53. (Original) The computer program product of claim 52, further comprising instructions for sending the second reply to the requester in the distributed data processing system.

54. (Currently Amended) A computer program product in a computer readable medium for retrieving object data from a repository, comprising:

instructions for receiving a first query for the object data from a requester in the distributed data processing system, wherein the first query is in a protocol recognized by an OID abstraction layer; wherein the OID abstraction layer is capable of receiving queries for objects in two or more different protocols and supports the two or more different protocols by mapping queries from multiple protocol interfaces to application program interface (API) requests that the repository understands;

instructions for interpreting the first query according to the protocol recognized by the OID abstraction layer, wherein the protocol recognized by the OID abstraction layer is one of the two or more different protocols;

instructions for locating a repository that contains the object data requested in the first query based on a registry associated with the OID abstraction layer; and

instructions for retrieving the object data from the repository using an OID abstraction layer application program interface (API) ~~The computer program product of claim 47~~, wherein each repository in a plurality repositories contains Object Identifier (OID) tree structures, and wherein the plurality of repositories are formatted to support the two or more different protocols.

55. (Currently Amended) The computer program product of claim [[47]] 54, wherein instructions for receiving a first query recognize a Simple Network Management Protocol (SNMP) query.

56. (Currently Amended) The computer program product of claim [[47]] 54, wherein instructions for receiving a first query recognize a Lightweight Directory Access Protocol (LDAP) query.

57. (Currently Amended) The computer program product of claim [[47]] 54, wherein instructions for receiving a first query recognize a Common Information Model used in conjunction with eXtendable Markup Language (CIM/XML) query.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to LeChi Truong whose telephone number is (571) 272 3767. The examiner can normally be reached on 8 - 5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomson, William can be reached on (571) 272 3718. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR of Public PAIP. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIP system, contact the Electronic Business Center (EBC) at 866-217-9197(toll-free).

LeChi Truong

May 8, 2007

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Allowable Subject Matter

2. Claims 1, 3, 4, 5, 6, 8-13, 16, 18-24 are allowed.

3. The following is an examiner's statement of reasons for allowance:

As to claims 2-4, 6, 8, 11, 14-19, 21-23, 25, 27, 30, 33-38, 40-42, 44, 46,49, 52-57, the prior art as taught by Spofford et al (US. 5,913037), Dobbins et al (US. Patent 5,951649) and Pearson (US. Patent 6,023684) do not teach on render obvious the limitations recited in claims 6, 8, 14, 16, 25, 27, 33, 35, 44, 46, 52, 54, when taken in the context of the claims as a whole, adding the OID tree structure to a repository associated with the OID abstraction layer, wherein the repository is configured such that the repository recognizes requests from an application program interface (API) associated with the OID abstraction layer and sends reply messages to the API containing information retrieved from the repository, wherein the OID abstraction layer receives the information retrieved from the repository through the API and encapsulates the information in a reply message to a target protocol interface, wherein the reply message is formatted for an appropriate protocol for the target protocol interface, and wherein the appropriate protocol is one of the two or more different protocols as recited in the independent claims 6, 8, 14, 16, 25, 27, 33, 35, 44, 46, 52, 54 . Moreover, evidence for modifying the prior art teachings by one of ordinary skill level in the art was not uncovered so as to result in the invention as recited in claims 6, 8, 14, 16, 25, 27, 33, 35, 44, 46, 52, and 54.

4. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to LeChi Truong whose telephone number is (571) 272 3767. The examiner can normally be reached on 8 - 5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomson, William can be reached on (571) 272 3718. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR of Public PAIP. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIP system, contact the Electronic Business Center (EBC) at 866-217-9197(toll-free).

LeChi Truong

May 8, 2007

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER